REMARKS

Claims 1-13, 15-25 and 37 remain pending in the application. Claims 1, 11, 12, and 21-23 have been amended without introduction of new matter. Favorable reconsideration is respectfully requested in view of the above amendments and the following remarks.

Claim 12 has been amended merely to correct a grammatical error ("comprising" changed to "comprises"). This amendment has not resulted in any change in scope of coverage.

Claims 11-13 and 37 stand rejected under 35 U.S.C. § 112, second paragraph, as allegedly being indefinite. In particular, the Office observes that in claim 11, "said error message" lacks antecedent basis. In response, claim 11 has been amended to now recite "an error message." As claims 12-13 and 37 depend from claim 11, no further amendments are required to address the rejection. Accordingly, it is respectfully requested that the rejection of claims 11-13 and 37 under the second paragraph of 35 U.S.C. § 112 be withdrawn.

Claims 1-13, 15-25, and 37 again stand rejected under 35 U.S.C. § 103(a) as allegedly being unpatentable over Greene (U.S. Patent No. 6,631,419 – henceforth "Greene") in view of Wilkinson III et al. (U.S. Patent No. 6,014,659 – henceforth "Wilkinson"). This rejection is respectfully traversed.

The Greene and Wilkinson documents have figured consistently throughout the prosecution of this application and, on various occasions, both Applicant and the Examiner have expressed their opinions on the relevance or otherwise of the disclosure in, and teaching of, Greene and Wilkinson, singly or in combination. It is nevertheless still worth a reminder of the disclosure in Greene.

The Greene patent describes a system in which data are stored in locations in a memory array 108 (Figure 1, Greene). The memory is subdivided into discrete regions 116, 118, 120. These regions are variously stated in Greene to be "independent memory arrays" (Column 4, lines 52-54) and "logically independent", in that they are separately addressed (Column 8, lines15-16). Moreover, in order to access a particular address within the array 118, the incoming key DEST_IP at 102 (Figure 1, Greene) is split into three components L1, L2, L3. The first component L1 interrogates the first array region 116 at 122-1 and returns either a result or a pointer D1 at 124-1 to the second region 118. The pointer is combined in A2 GEN 112 (Figure 1, Greene) with the second part L2 of the input key to produce a second level address A2 to access the second region 118. The procedure is reiterated to generate (at

114) from a pointer D2 returned from the second region 118, in combination with the third part L3 of the incoming address key, a third level address A3 to access the third region 120 of the array 108. This procedure is described in Column 7, line 6 to Column 7, line 12 with reference to Figure 1. The overall process is described as "pipelined" in Column 4, line 53.

As the Office rightly states, Greene does not disclose or teach a lookup engine comprising a plurality of lookup state machines connected in parallel to enable multiple lookups to be carried out concurrently. Quite the contrary, there is clearly a need for all three stages of the memory array to be visited ***sequentially*** in order to access data stored in the array, as is inherent in the array being "pipelined", as just noted in the above paragraph.

As previously discussed, Wilkinson discloses a trie searching procedure in which nodes are selectively eliminated in order to reduce memory storage. When a node is encountered that (i) has a single "child" node dependent from it and (ii) contains no result matching the particular segment of the search request/argument or has the same result as the parent, that node is eliminated from the search path. If so, a digit string is established that contains all of the segments of the eliminated node or nodes. Each digit of the string is compared against each segment of the search argument and the search proceeds only if all segments match, otherwise the search terminates at the node preceding the digit string. It is clear that Wilkinson is only capable of handling a *single* request at any one time.

The Office has cited Wilkinson on the basis that it discloses a searching procedure employing multiple state machines, thereby alleging that it would have been obvious to one skilled in the art at the time of Applicant's invention, to have combined the teaching of Wilkinson with that of Greene. In response, it is respectfully asserted that, in view of the fact that Wilkinson does indeed use multiple state machines but is still incapable of searching more than a single request at a time, it is manifestly not the case that merely providing a plurality of state machines accessing the same memory at the same time would necessarily achieve the stated object of Applicant's invention, namely of *concurrently* conducting a plurality of *different* requests for *different* items of information/data in the same memory by utilizing a plurality of parallel paths and state machines in order to complete a set or "batch" of multiple requests in the shortest time possible. This means not only performing the accesses **in parallel** rather than sequentially, as in Greene, but also allowing accesses to "leap frog" as necessary (see further discussion below). It would be quite pointless to adopt the Greene *sequential* approach and to modify it by adding *parallel*

processors since each processor would have to wait for a current search string to complete before the next could access the first of the sequential arrays.

As also previously noted, Applicant's multiple state machines are used to accelerate the *overall* look-up process, where the ability to look up several values in parallel means that *all* of the look-ups can be completed faster, that is, the latency for any given batch of search requests is reduced. The concurrent look-ups in the present invention can take different amounts of time and can emerge in a different order to the original request ("leap-frogging"). In Greene, lookups must take place in the same order as their arrival order since each search has to be conducted sequentially through all the regions of the memory array: there is no scope for "leap-frogging". In Applicant's invention, all of the finite state machines are able to reference the whole of the lookup table data at the same time whereas in Greene the state machine can only reference one region of the array at a time. It is conceivable, although not explicitly contemplated, that the prior art systems *may* be able to perform concurrent multiple lookups in a memory array but they do so in different lookup tables or regions of the array. There is never a time when the Greene state machine looks up more than one reference in response to more than one request in *all* areas of the memory array at the same time. Moreover, the fact that Wilkinson, despite having more than one state machine, is still incapable of performing more than one request at a time, reinforces Applicant's view that combining the teaching of Wilkinson with the architecture of Greene would not be obvious and would not in any case lead to the claimed invention for the reasons set out above.

Clearly, there would be absolutely no motivation for the person of ordinary skill in the art even to contemplate combining the teaching of Wilkinson with that of Greene because the combination would not achieve the desired objective and would not guarantee that the processors would be able to access the whole of the data memory in parallel, despite there being parallel processors (i.e., there would not be the requisite likelihood of success).

> As stated on page 3 at lines 17-24 of Applicant's specification:
>
> Each look up state machine accesses storage means, preferably comprising a plurality of parallel, independent memory banks, in which the look up table may be constructed on the basis of a trie, more preferably a Patricia tree structure. Such a look up table provides increased performance by doing *multiple parallel lookups to multiple memory banks in parallel.*

(Emphasis added.)

Page 13, lines 1-2 also confirm that "Multiple client blocks can submit lookup requests *simultaneously*" (emphasis added). Again, it is stated in page 14, lines 12-16 with reference to Figure 2 that "The distributor block 204 watches the lookup state machines 206a, 206b, 206c, 206d and sends a key to any one that is available to do a new lookup. A priority encoder may be used to choose the first ready state machine." And yet again, in the text spanning line 37 of page 16 through line 1 of page 17, "...there are enough state machines so that one of them is doing a memory access on nearly every cycle...."

It is clear that, in Applicant's invention, there is no necessity for one search to complete before another can commence. All that is needed is for a state machine to be available to perform a new search while other state machines continue with their already commenced searches. Since all state machines have access to all the memory banks all the time via the distributor block 204, multiple independent lookups can be performed concurrently, with each state machine having access to all the data in the whole of the memory.

As is settled law, an item of prior art cannot simply be collocated with another to render a claim obvious without motive or incentive. In the present circumstances, it is respectfully submitted that there is neither motive nor incentive to combine Wilkinson with Greene in the manner proposed by the Office. It is therefore respectfully submitted that Wilkinson does not provide the missing link required to take the person of ordinary skill in the art from Greene to the Applicant's claimed invention.

Despite Applicant's view as expressed above, claims 1, 21, 22 and 23 have nevertheless been amended to define, and thereby emphasize, that "the state machines all hav[e] concurrent access to the entries in the whole of the look up table whenever they perform a look up." (See the passage quoted above from page 3 of Applicant's specification). The importance of this feature must now be clear from the above discussion: it permits each of the state machines, once it becomes available after completing a previous search, to access the whole of the memory array. In contrast, Greene carries out a search in sequential parts of the memory array, so it is impossible for one state machine to attempt to access a part of the memory that is already being searched by another state machine in parallel. This is what is really meant by the word "concurrent" in claim 1 but the present amendment should clarify the intention and make a clearer divide between Applicant's invention and the cited references.

This amendment also answers any question as to whether regions of a table constitute tables in their own right or whether such regions are parts of a single table. Provided all the state machines in the look up engine have access to the whole of the table whenever they are performing a search, they satisfy the terms of Applicant's amended claims, which the cited prior art references do not.

For at least the foregoing reasons, it is respectfully asserted that independent claims 1, 21, 22, and 23, as well as their dependent claims 2-13, 15-20, 24-25 and 37 are patentably distinguishable over the prior art of record. Therefore, it is respectfully requested that the rejection of these claims under 35 U.S.C. §103(a) be withdrawn.

The application is believed to be in condition for allowance. Prompt notice of same is respectfully requested.

Respectfully submitted,

Potomac Patent Group PLLC

Date: July 26, 2006

By: _____
Kenneth B. Leffler
Registration No. 36,075

P.O. Box 270
Fredericksburg, Virginia 22404
703-718-8884